Project on finding the more accurate method among Taylor series expansion, Eular Method and the 4th Order Runge Kutta Method

Group Members

Muhaimenur Rahman
Reg.no. : 2017132009
Ankan Das
Reg.no.: 2017132021
Rajan Deb Joyanto
Reg.no.: 2017132033
Abdul Ahad Sumon
Reg.no.: 2017132074

Abstract:

This project is to find out the more accurate numerical value among Taylor series expansion, Euler method and 4th order Runge Kutta method for decay equation. We find that Taylor series gives more accurate value at some certain values, it also depends on the terms is taken. But we want to take one method in overall then we can say 4th order Runge Kutta method is the most accurate among this three in overall.

Introduction:

This paper is dedicated to a comprehensive study of accuracy of several numerical methods used for solving the decay equation by inspecting the respective errors computed during the process. We have the method of Taylor series expansion, Eular Method and the 4th Order Runge Kutta Method to numerically solve for a very important differential equation, namely the decay equation, which is confronted quiet frequently in science and engineering. Even though the methods vary in their approximation to find the accurate numerical solution, but they all have same objective: finding the best solution numerically.

Theory:

Decay equation:

Decay equation is most generally expressed as

$$N' = kN$$

Here, N' is the 1st derivative of the function N with respect to time(t) and k is the proportionality constant. The equation reduces to a decay equation provided the k is (-)ve.

The general solution,

$N = N_0 e^{kt}$

which is known, however is irrelevant to the process of numerically tracing the function N in any moment in time. The equation boiled down to a proportionality,

$N' \propto N$

exposes the underlying physical aspect of the process which is the decay rate depends on how much of the object to be decayed is existent to begin with

The initial value reads, $N(t_0) = N_0$

Taylor series expansion:

Any Ordinay Differential Equation of the form

$N' = f(t,N)$

with initial condition,

$N(t_0) = N_0$

can be expanded around the point x_0 by the following expansion,

$N(t) = N_0 + (t-t_0) N_0' + (t-t_0)^2 N_0'' /2! + (t-t_0)^3 N_0''' /3! + (t-t_0)^4 N_0'''' /4! + \ldots\ldots$

as such it is necessary to know,

$N_0', N_0'', N_0''', N_0'''', \ldots..$

Moreover, the more the number of terms we use from the expansion, the better our approximation is compared to the original solution/analytic solution.

Euler method:

Euler method gives the solution in the form of a set of tabulated values.

Suppose that we wish to solve the equation,

$N' = f(t,N)$

for the values of N at $t = t_r = t_0 + rh$ ($r = 1,2,\ldots..$). Integrating equation, we obtain

$N_1 = N_0 + \int_{t0}^{t1} f(t,N)dt$

Assuming that $f(t,N) = f(t_0,N_0)$ in $t_0 \leq t \leq t_1$, this gives Eular's formula

$N_1 = N_0 + hf(t_0,N_0)$

Similarly for the range $t_1 \leq t \leq t_2$, we have

$N_2 = N_1 + \int_{t1}^{t2} f(t,N)dt$

Substituting $f(t_1,N_1)$ for f(t.N) in $t_1 \leq t \leq t_2$ we obtain

$N_2 = N_1 + hf(t_1,N_1)$

Proceeding in this way, we obtain the general formula

$N_{n+1} = N_n + hf(t_n,N_n)$, n = 0,1,2,……..

The process is very slow and to obtain reasonable accuracy with Euler's method, we need to take a smaller value for h. Because of this restriction on h, the method is unsuitable for practical use and a modification of it, known as the modified Euler method, which gives more accurate results.

4th Order Runge Kutta method:

The Runge-Kutta method attempts to overcome the problem of the Euler's method, as far as the choice of a sufficiently small step size is concerned, to reach a reasonable accuracy in the problem resolution. For the differential equation

$N'=f(t,N)$

where $N(t_0) = N_0$ the Runge-Kutta of fourth-order method (RK4) method is defined using the following recursion formula:

$N_{n+1}=N_n+1/6(k_1+2\ k_2+2\ k_3+\ k_4)$ ……………(1)

where:

$k_1=hf(t_n,N_n)$

$k_2=hf(\ t_n+\frac{h}{2}\ ,\ N_n+\frac{k1}{2}\ )$

$k_3=\ hf(\ t_n+\frac{h}{2}\ ,\ N_n+\frac{k2}{2}\ )$

$k_4=h\ f(\ t_n+h\ ,\ N_n+k_3\ )$

h=step size

Runge-Kutta methods of any order can be derived, although the derivation of an order higher than four can become extremely complicated. The most popular method used is the RK4, as represented in Eq.(1)).

Data and Graph:

We are taking 3 examples to analysis and find out the most accurate method.

First we are taking

```
initial time t0 : 0
initial atom no : 1000
value of h : .01
value of constant : 3
how many terms of taylor series do you want to count : 10
```

The calculated value for this condition

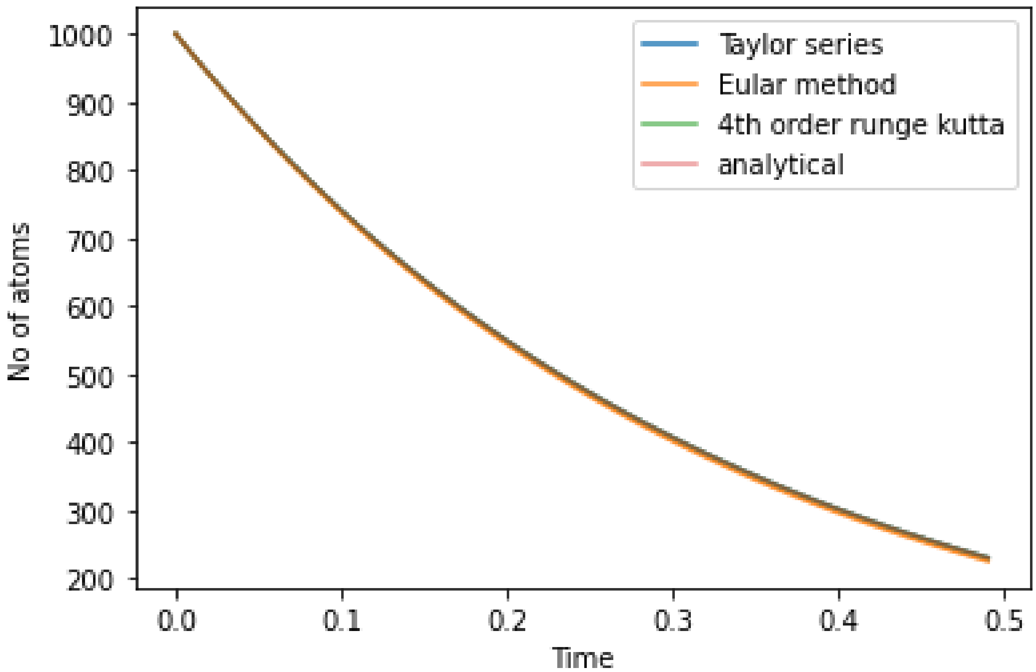
First 20 values:

	Time	Analytical value	Taylor series	Eular method	Runge Kutta method
0	0.00	1000.000000	1000.000000	1000.000000	1000.000000
1	0.01	970.445534	970.445534	970.000000	970.445534
2	0.02	941.764534	941.764534	940.900000	941.764534
3	0.03	913.931185	913.931185	912.673000	913.931186
4	0.04	886.920437	886.920437	885.292810	886.920437
5	0.05	860.707976	860.707976	858.734026	860.707977
6	0.06	835.270211	835.270211	832.972005	835.270212
7	0.07	810.584246	810.584246	807.982845	810.584247
8	0.08	786.627861	786.627861	783.743359	786.627862
9	0.09	763.379494	763.379494	760.231059	763.379496
10	0.10	740.818221	740.818221	737.424127	740.818222
11	0.11	718.923733	718.923733	715.301403	718.923735
12	0.12	697.676326	697.676326	693.842361	697.676328
13	0.13	677.056874	677.056874	673.027090	677.056876
14	0.14	657.046820	657.046820	652.836277	657.046822
15	0.15	637.628152	637.628152	633.251189	637.628154
16	0.16	618.783392	618.783392	614.253653	618.783394
17	0.17	600.495579	600.495579	595.826044	600.495581
18	0.18	582.748252	582.748252	577.951263	582.748255
19	0.19	565.525439	565.525439	560.612725	565.525441

Last 20 values:

	Time	Analytical value	Taylor series	Eular method	Runge Kutta method
30	0.30	406.569660	406.569667	401.007069	406.569662
31	0.31	394.553710	394.553721	388.976856	394.553713
32	0.32	382.892886	382.892901	377.307551	382.892889
33	0.33	371.576691	371.576712	365.988324	371.576694
34	0.34	360.594940	360.594969	355.008675	360.594943
35	0.35	349.937749	349.937788	344.358414	349.937752
36	0.36	339.595526	339.595579	334.027662	339.595528
37	0.37	329.558961	329.559033	324.006832	329.558964
38	0.38	319.819022	319.819118	314.286627	319.819024
39	0.39	310.366941	310.367070	304.858028	310.366944
40	0.40	301.194212	301.194381	295.712287	301.194214
41	0.41	292.292578	292.292799	286.840919	292.292580
42	0.42	283.654026	283.654314	278.235691	283.654029
43	0.43	275.270783	275.271155	269.888620	275.270786
44	0.44	267.135302	267.135780	261.791962	267.135304
45	0.45	259.240261	259.240871	253.938203	259.240263
46	0.46	251.578553	251.579329	246.320057	251.578555
47	0.47	244.143283	244.144264	238.930455	244.143286
48	0.48	236.927759	236.928992	231.762542	236.927761
49	0.49	229.925485	229.927030	224.809665	229.925488

Graph:



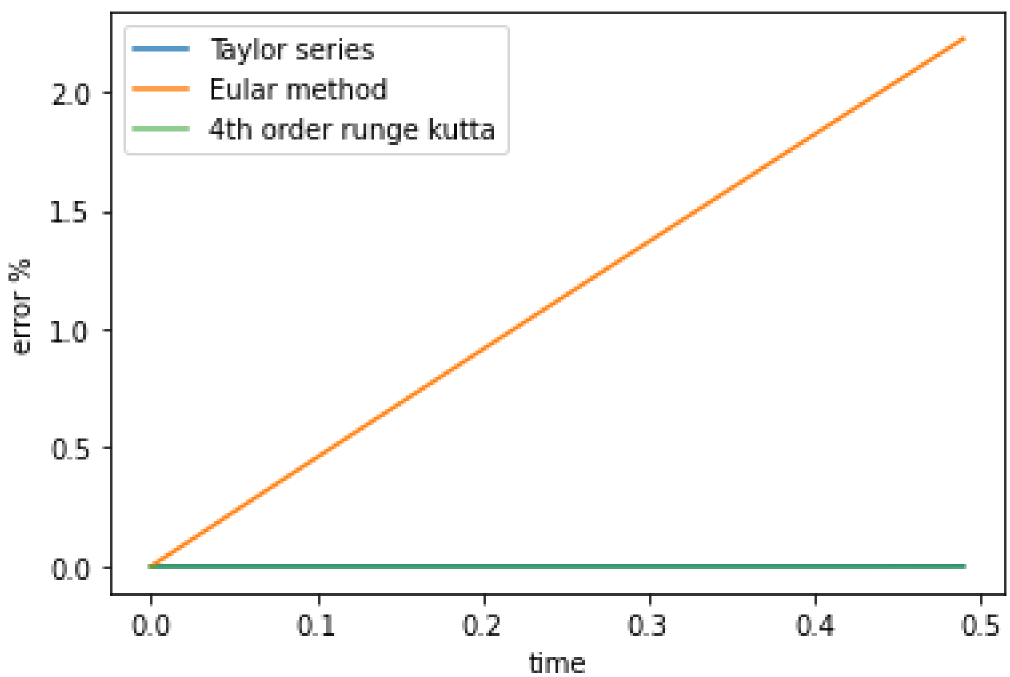
Error value of first 20 :

	Time	Taylor series error	Eular method error	Runge Kutta error
0	0.00	0.000000e+00	0.000000	0.000000e+00
1	0.01	0.000000e+00	0.045910	2.076282e-08
2	0.02	0.000000e+00	0.091799	4.152563e-08
3	0.03	0.000000e+00	0.137667	6.228845e-08
4	0.04	1.281816e-14	0.183514	8.305126e-08
5	0.05	1.320853e-14	0.229340	1.038141e-07
6	0.06	2.722157e-14	0.275145	1.245769e-07
7	0.07	8.415177e-14	0.320929	1.453397e-07
8	0.08	4.624777e-13	0.366692	1.661025e-07
9	0.09	1.802001e-12	0.412434	1.868654e-07
10	0.10	5.816179e-12	0.458155	2.076282e-07
11	0.11	1.714181e-11	0.503855	2.283910e-07
12	0.12	4.590321e-11	0.549533	2.491538e-07
13	0.13	1.137780e-10	0.595191	2.699166e-07
14	0.14	2.642468e-10	0.640828	2.906794e-07
15	0.15	5.801945e-10	0.686444	3.114422e-07
16	0.16	1.213072e-09	0.732039	3.322051e-07
17	0.17	2.429243e-09	0.777614	3.529679e-07
18	0.18	4.682917e-09	0.823167	3.737307e-07
19	0.19	8.725550e-09	0.868699	3.944935e-07

Error value of last 20 :

	Time	Taylor series error	Eular method error	Runge Kutta error
30	0.30	0.000002	1.368177	6.228845e-07
31	0.31	0.000003	1.413459	6.436473e-07
32	0.32	0.000004	1.458720	6.644101e-07
33	0.33	0.000006	1.503961	6.851729e-07
34	0.34	0.000008	1.549180	7.059357e-07
35	0.35	0.000011	1.594379	7.266985e-07
36	0.36	0.000016	1.639557	7.474614e-07
37	0.37	0.000022	1.684715	7.682242e-07
38	0.38	0.000030	1.729852	7.889870e-07
39	0.39	0.000041	1.774968	8.097498e-07
40	0.40	0.000056	1.820063	8.305126e-07
41	0.41	0.000076	1.865138	8.512754e-07
42	0.42	0.000101	1.910192	8.720383e-07
43	0.43	0.000135	1.955225	8.928011e-07
44	0.44	0.000179	2.000237	9.135639e-07
45	0.45	0.000236	2.045229	9.343267e-07
46	0.46	0.000308	2.090200	9.550895e-07
47	0.47	0.000402	2.135151	9.758523e-07
48	0.48	0.000521	2.180081	9.966151e-07
49	0.49	0.000672	2.224990	1.017378e-06

Error graph:



Now 2nd example:

```
initial time t0 : 0
initial atom no : 1000
value of h : .01
value of constant : 3
how many terms of taylor series do you want to count : 5
```

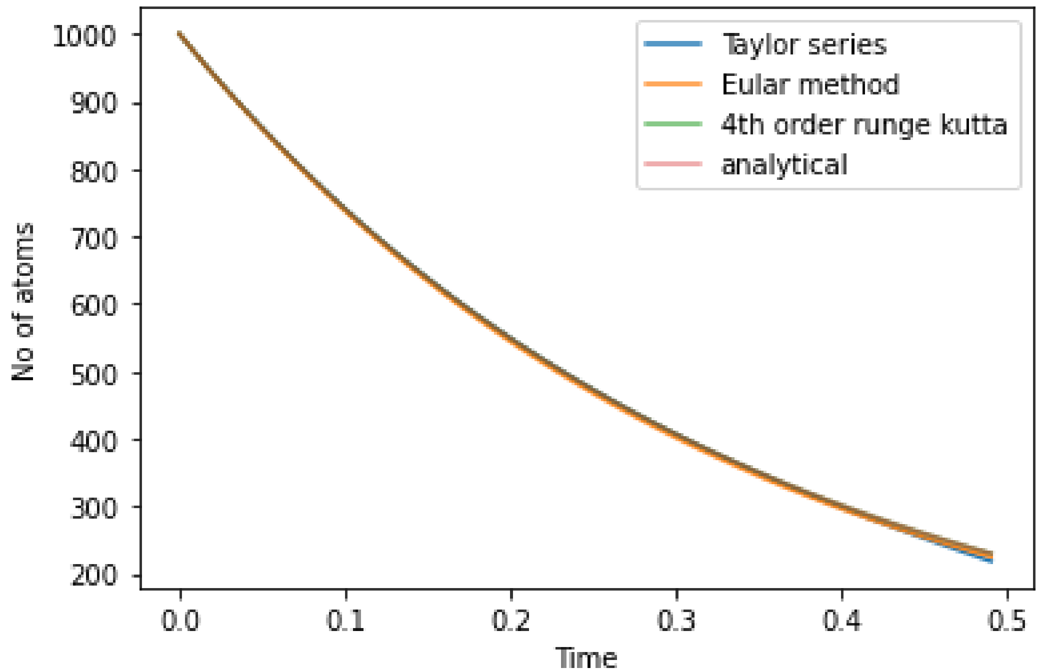
First 20 values:

	Time	Analytical value	Taylor series	Eular method	Runge Kutta method
0	0.00	1000.000000	1000.000000	1000.000000	1000.000000
1	0.01	970.445534	970.445534	970.000000	970.445534
2	0.02	941.764534	941.764534	940.900000	941.764534
3	0.03	913.931185	913.931185	912.673000	913.931186
4	0.04	886.920437	886.920433	885.292810	886.920437
5	0.05	860.707976	860.707961	858.734026	860.707977
6	0.06	835.270211	835.270165	832.972005	835.270212
7	0.07	810.584246	810.584130	807.982845	810.584247
8	0.08	786.627861	786.627604	783.743359	786.627862
9	0.09	763.379494	763.378976	760.231059	763.379496
10	0.10	740.818221	740.817250	737.424127	740.818222
11	0.11	718.923733	718.922021	715.301403	718.923735
12	0.12	697.676326	697.673452	693.842361	697.676328
13	0.13	677.056874	677.052247	673.027090	677.056876
14	0.14	657.046820	657.039631	652.836277	657.046822
15	0.15	637.628152	637.617320	633.251189	637.628154
16	0.16	618.783392	618.767503	614.253653	618.783394
17	0.17	600.495579	600.472813	595.826044	600.495581
18	0.18	582.748252	582.716302	577.951263	582.748255
19	0.19	565.525439	565.481424	560.612725	565.525441

Last 20 values:

	Time	Analytical value	Taylor series	Eular method	Runge Kutta method
30	0.30	406.569660	405.916750	401.007069	406.569662
31	0.31	394.553710	393.761931	388.976856	394.553713
32	0.32	382.892886	381.938668	377.307551	382.892889
33	0.33	371.576691	370.433417	365.988324	371.576694
34	0.34	360.594940	359.232667	355.008675	360.594943
35	0.35	349.937749	348.322914	344.358414	349.937752
36	0.36	339.595526	337.690639	334.027662	339.595528
37	0.37	329.558961	327.322282	324.006832	329.558964
38	0.38	319.819022	317.204218	314.286627	319.819024
39	0.39	310.366941	307.322733	304.858028	310.366944
40	0.40	301.194212	297.664000	295.712287	301.194214
41	0.41	292.292578	288.214053	286.840919	292.292580
42	0.42	283.654026	278.958766	278.235691	283.654029
43	0.43	275.270783	269.883824	269.888620	275.270786
44	0.44	267.135302	260.974705	261.791962	267.135304
45	0.45	259.240261	252.216648	253.938203	259.240263
46	0.46	251.578553	243.594637	246.320057	251.578555
47	0.47	244.143283	235.093370	238.930455	244.143286
48	0.48	236.927759	226.697236	231.762542	236.927761
49	0.49	229.925485	218.390296	224.809665	229.925488

Graph:



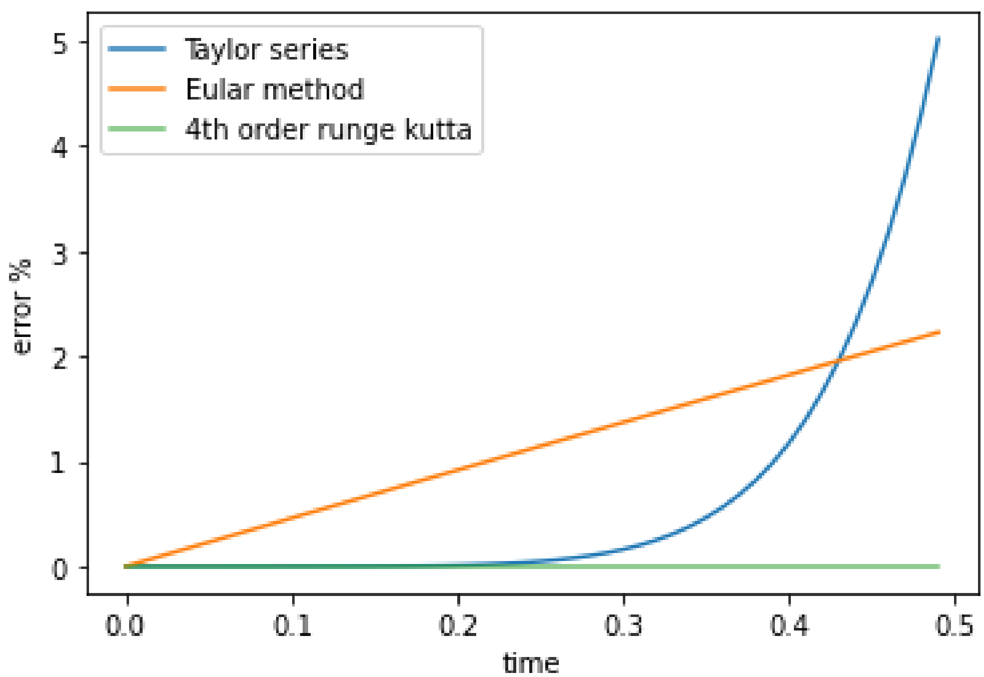
Error values of first 20:

	Time	Taylor series error	Eular method error	Runge Kutta error
0	0.00	0.000000e+00	0.000000	0.000000e+00
1	0.01	1.038878e-10	0.045910	2.076282e-08
2	0.02	6.822167e-09	0.091799	4.152563e-08
3	0.03	7.973557e-08	0.137667	6.228845e-08
4	0.04	4.596982e-07	0.183514	8.305126e-08
5	0.05	1.799398e-06	0.229340	1.038141e-07
6	0.06	5.513338e-06	0.275145	1.245769e-07
7	0.07	1.426597e-05	0.320929	1.453397e-07
8	0.08	3.261854e-05	0.366692	1.661025e-07
9	0.09	6.785738e-05	0.412434	1.868654e-07
10	0.10	1.310283e-04	0.458155	2.076282e-07
11	0.11	2.382046e-04	0.503855	2.283910e-07
12	0.12	4.120179e-04	0.549533	2.491538e-07
13	0.13	6.834848e-04	0.595191	2.699166e-07
14	0.14	1.094165e-03	0.640828	2.906794e-07
15	0.15	1.698687e-03	0.686444	3.114422e-07
16	0.16	2.567691e-03	0.732039	3.322051e-07
17	0.17	3.791219e-03	0.777614	3.529679e-07
18	0.18	5.482624e-03	0.823167	3.737307e-07
19	0.19	7.783027e-03	0.868699	3.944935e-07

Error values of last 20:

	Time	Taylor series error	Eular method error	Runge Kutta error
30	0.30	0.160590	1.368177	6.228845e-07
31	0.31	0.200677	1.413459	6.436473e-07
32	0.32	0.249213	1.458720	6.644101e-07
33	0.33	0.307682	1.503961	6.851729e-07
34	0.34	0.377785	1.549180	7.059357e-07
35	0.35	0.461464	1.594379	7.266985e-07
36	0.36	0.560928	1.639557	7.474614e-07
37	0.37	0.678688	1.684715	7.682242e-07
38	0.38	0.817588	1.729852	7.889870e-07
39	0.39	0.980842	1.774968	8.097498e-07
40	0.40	1.172072	1.820063	8.305126e-07
41	0.41	1.395357	1.865138	8.512754e-07
42	0.42	1.655277	1.910192	8.720383e-07
43	0.43	1.956967	1.955225	8.928011e-07
44	0.44	2.306171	2.000237	9.135639e-07
45	0.45	2.709306	2.045229	9.343267e-07
46	0.46	3.173528	2.090200	9.550895e-07
47	0.47	3.706804	2.135151	9.758523e-07
48	0.48	4.317992	2.180081	9.966151e-07
49	0.49	5.016925	2.224990	1.017378e-06

Error graph:



3rd example:

```
initial time t0 : 0
initial atom no : 1000
value of h : .1
value of constant : 3
how many terms of taylor series do you want to count : 5
```

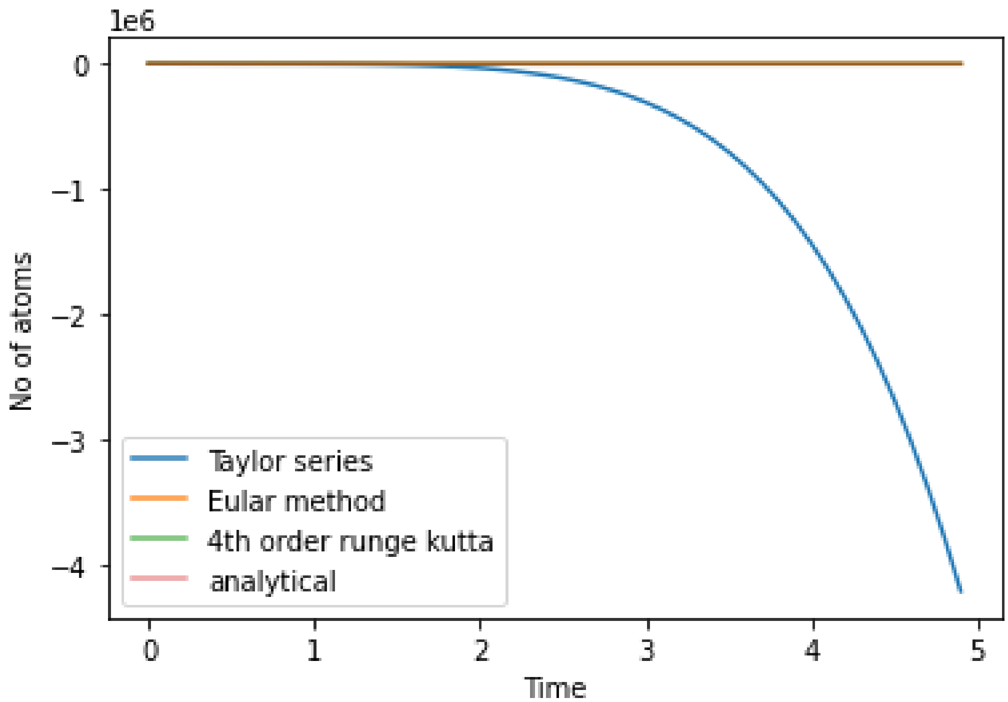
First 20 values:

	Time	Analytical value	Taylor series	Eular method	Runge Kutta method
0	0.0	1000.000000	1000.00000	1000.000000	1000.000000
1	0.1	740.818221	740.81725	700.000000	740.837500
2	0.2	548.811636	548.75200	490.000000	548.840201
3	0.3	406.569660	405.91675	343.000000	406.601403
4	0.4	301.194212	297.66400	240.100000	301.225567
5	0.5	223.130160	210.15625	168.070000	223.159196
6	0.6	165.298888	127.93600	117.649000	165.324701
7	0.7	122.456428	31.49575	82.354300	122.478738
8	0.8	90.717953	-105.15200	57.648010	90.736842
9	0.9	67.205513	-316.90475	40.353607	67.221255
10	1.0	49.787068	-650.00000	28.247525	49.800027
11	1.1	36.883167	-1164.44525	19.773267	36.893727
12	1.2	27.323722	-1936.44800	13.841287	27.332257
13	1.3	20.241911	-3060.84575	9.688901	20.248761
14	1.4	14.995577	-4653.53600	6.782231	15.001041
15	1.5	11.108997	-6853.90625	4.747562	11.113334
16	1.6	8.229747	-9827.26400	3.323293	8.233175
17	1.7	6.096747	-13767.26675	2.326305	6.099444
18	1.8	4.516581	-18898.35200	1.628414	4.518697
19	1.9	3.345965	-25478.16725	1.139890	3.347620

Last 20 values:

	Time	Analytical value	Taylor series	Eular method	Runge Kutta method
30	3.0	0.123410	-3.077000e+05	0.022539	0.123506
31	3.1	0.091424	-3.671665e+05	0.015778	0.091498
32	3.2	0.067729	-4.355588e+05	0.011044	0.067785
33	3.3	0.050175	-5.138549e+05	0.007731	0.050218
34	3.4	0.037170	-6.031019e+05	0.005412	0.037203
35	3.5	0.027536	-7.044195e+05	0.003788	0.027562
36	3.6	0.020400	-8.190017e+05	0.002652	0.020419
37	3.7	0.015112	-9.481193e+05	0.001856	0.015127
38	3.8	0.011195	-1.093123e+06	0.001299	0.011207
39	3.9	0.008294	-1.255444e+06	0.000910	0.008302
40	4.0	0.006144	-1.436600e+06	0.000637	0.006151
41	4.1	0.004552	-1.638193e+06	0.000446	0.004557
42	4.2	0.003372	-1.861916e+06	0.000312	0.003376
43	4.3	0.002498	-2.109552e+06	0.000218	0.002501
44	4.4	0.001851	-2.382979e+06	0.000153	0.001853
45	4.5	0.001371	-2.684171e+06	0.000107	0.001373
46	4.6	0.001016	-3.015201e+06	0.000075	0.001017
47	4.7	0.000752	-3.378243e+06	0.000052	0.000753
48	4.8	0.000557	-3.775574e+06	0.000037	0.000558
49	4.9	0.000413	-4.209579e+06	0.000026	0.000413

Graph:



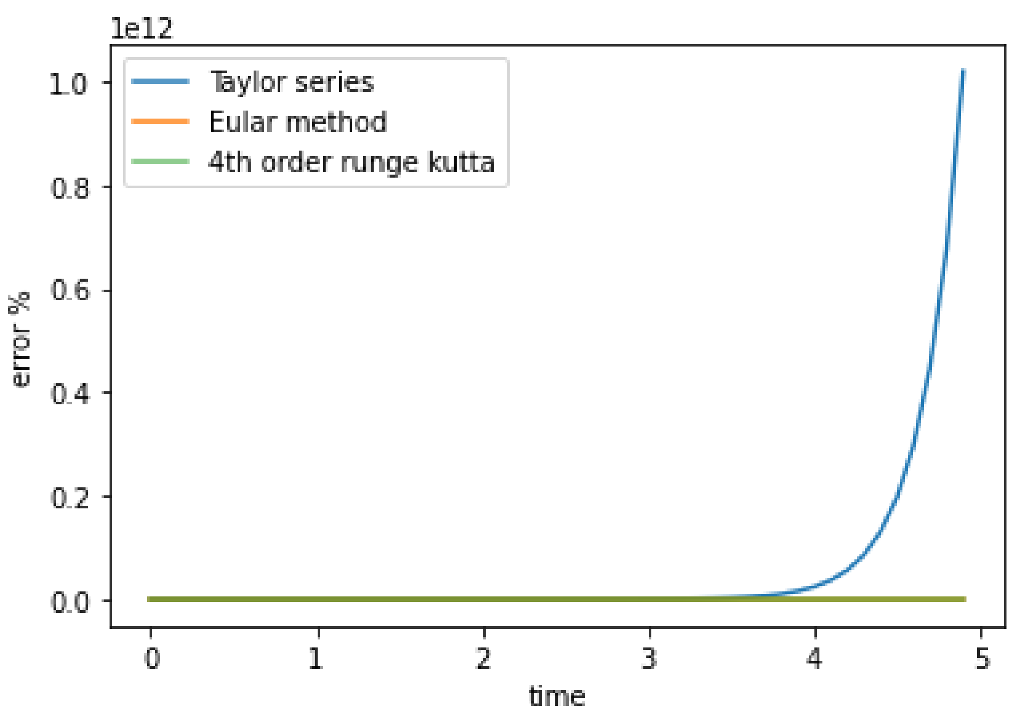
Error values of first 20:

	Time	Taylor series error	Eular method error	Runge Kutta error
0	0.0	0.000000	0.000000	0.000000
1	0.1	0.000131	5.509883	0.002602
2	0.2	0.010866	10.716179	0.005205
3	0.3	0.160590	15.635613	0.007808
4	0.4	1.172072	20.283993	0.010410
5	0.5	5.814503	24.676252	0.013013
6	0.6	22.603230	28.826503	0.015616
7	0.7	74.280035	32.748079	0.018218
8	0.8	215.910904	36.453582	0.020821
9	0.9	571.545766	39.954915	0.023424
10	1.0	1405.559900	43.263330	0.026027
11	1.1	3257.118361	46.389454	0.028631
12	1.2	7187.057789	49.343333	0.031234
13	1.3	15221.327638	52.134456	0.033837
14	1.4	31132.724221	54.771792	0.036440
15	1.5	61796.897883	57.263813	0.039044
16	1.6	119511.495171	59.618527	0.041647
17	1.7	225913.335064	61.843499	0.044251
18	1.8	418521.638849	63.945878	0.046854
19	1.9	761559.362741	65.932418	0.049458

Error values of last 20:

	Time	Taylor series error	Eular method error	Runge Kutta error
30	3.0	2.493320e+08	81.736183	0.078103
31	3.1	4.016075e+08	82.742498	0.080707
32	3.2	6.430932e+08	83.693367	0.083312
33	3.3	1.024132e+09	84.591843	0.085916
34	3.4	1.622536e+09	85.440815	0.088521
35	3.5	2.558135e+09	86.243009	0.091126
36	3.6	4.014812e+09	87.001003	0.093730
37	3.7	6.273815e+09	87.717233	0.096335
38	3.8	9.763961e+09	88.393999	0.098940
39	3.9	1.513711e+10	89.033476	0.101545
40	4.0	2.338135e+10	89.637719	0.104150
41	4.1	3.599045e+10	90.208668	0.106755
42	4.2	5.521672e+10	90.748159	0.109361
43	4.3	8.444794e+10	91.257925	0.111966
44	4.4	1.287678e+11	91.739603	0.114571
45	4.5	1.957878e+11	92.194741	0.117177
46	4.6	2.968794e+11	92.624802	0.119782
47	4.7	4.489966e+11	93.031167	0.122388
48	4.8	6.773662e+11	93.415141	0.124993
49	4.9	1.019454e+12	93.777959	0.127599

Error graph:



Analysis:

If we notice first two examples , we find that the difference between is only on taking the expansion terms of taylor expansion series and this. In first example we take 10 terms and in the 2nd example we take 5 terms, but if we look at the results we can see, in the first example error in taylor series expansion is approximately 0 but in the second example we can see a significant error as time going, as there is no change in h, there is no error change in Euler Method and 4^{th} Order Runge Kutta Method. For the first two examples, we can see 4^{th} Order Runge Kutta Method is more accurate then Euler Method. Now if we compare between Taylor series and Runge Kutta Method we can say Runge Kutta is more accurate than Taylor series, though in the first 20 values we see Taylor is giving more accurate value but as time increase we can see Runge Kutta Method giving more accurate values. And in the 2^{nd} example's graph we can clearly see that Runge Kutta is the most accurate method among this three method. In the third example, we increase the value of h, and as a result in Euler Method and Runge Kutta Method, error percentage is also increase. We can also see that at a certain value Taylor value is giving us approximately accurate value but after passing this certain value its giving us a huge error. In the 3^{rd} example we can see, 4^{th} Order Runge Kutta Method is giving us the more accurate value.

So, we can conclude that from this three examples, 4^{th} Order Runge Kutta Method gives us the most accurate value.